ELSEVIER
European Journal of Operational Research 139 (2002) 26–41
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Continuous Optimization

A weight set decomposition algorithm for finding all efficient extreme points in the outcome set of a multiple objective linear program

Harold P. Benson *, Erjiang Sun

Department of Decision and Information Sciences, University of Florida, Gainesville, FL 32611, USA

Received 21 July 2000; accepted 20 March 2001

Abstract

Various computational difficulties arise in using decision set-based vector maximization methods to solve multiple objective linear programming problems. As a result, several researchers have begun to explore the possibility of solving these problems by examining subsets of their outcome sets, rather than of their decision sets. In this article, we present and validate a basic weight set decomposition approach for generating the set of all efficient extreme points in the outcome set of a multiple objective linear program. Based upon this approach, we then develop an algorithm, called the Weight Set Decomposition Algorithm, for generating this set. A sample problem is solved using this algorithm, and the main potential computational and practical advantages of the algorithm are indicated.

Keywords: Multicriteria analysis; Multiple objective linear programming; Efficient point; Outcome set

1. Introduction

A multiple objective linear programming problem may be written

(MOLP)

$$\begin{array}{ll} \text{Vmax} & Cx \\ \text{s.t.} & x \in X, \end{array}$$

where C is a $p \times n$ matrix with $p \geqslant 2$ whose rows $c_i,\ i = 1, 2, \ldots, p$, are the coefficients of p linear objective functions $\langle c_i, x\rangle,\ i = 1, 2, \ldots, p$, and $X \subseteq \mathbb{R}^n$ is a nonempty polyhedral *decision set*. Throughout this article, we will assume that the decision set X is given by

$$X = \{x \in \mathbb{R}^n \,|\, Ax \leqslant b,\ x \geqslant 0\},$$

where A is an $m \times n$ matrix and $b \in \mathbb{R}^m$, and that X is compact. The *outcome set* Y for problem (MOLP) is

$$Y = \{Cx \,|\, x \in X\}.$$

From [30], Y is also a nonempty, compact polyhedron.

* Corresponding author. Tel.: +1-352-392-0134; fax: +1-352-392-5438.
E-mail address: benson@dale.cba.ufl.edu (H.P. Benson).

PII: S0377-2217(01)00153-9

Problem (MOLP) is one of the most popular models used in multiple criteria decision making. Numerous studies and applications of multicriteria problems and problem (MOLP) have been reported in the literature in literally hundreds of books, monographs, articles and chapters in books. For an overview of these studies and applications, see, for instance [1,11,12,18,25–28,31, 32,34,36,37,39,40] and references therein.

Many approaches for analyzing and solving problem (MOLP) use the concept of efficiency. A point $x^0 \in \mathbb{R}^n$ is called an *efficient solution* for problem (MOLP) when $x^0 \in X$ and there exists no point $x \in X$ such that $Cx \geqslant Cx^0$ and $Cx \neq Cx^0$. Similarly, a point $y^0 \in \mathbb{R}^p$ is called an *efficient outcome* for problem (MOLP) when $y^0 \in Y$ and there exists no $y \in Y$ such that $y \geqslant y^0$ and $y \neq y^0$. The set of all efficient solutions and the set of all efficient outcomes for problem (MOLP) are called the *efficient decision set* and the *efficient outcome set*, respectively, for the problem, and are denoted X_E and Y_E, respectively, in this article.

One of the most common solution approaches for problem (MOLP) that relies upon the efficiency concept is called the *vector maximization* approach. In the most common forms of this approach, first either the entire efficient decision set X_E is generated or a representative subset of X_E is generated. Often, the representative subset generated is $X_E \cap X_{ex}$ where X_{ex} denotes the set of all extreme points of X. Next, the generated set is presented to the decision maker, who then chooses a most preferred solution for problem (MOLP) from the set.

Under mild assumptions, it is easy to show that any most preferred solution of the decision maker for problem (MOLP) must be an efficient solution for the problem. One of the main attractions of the vector maximization approach is that it restricts the search for a most preferred solution to X_E. Furthermore, it generates all of X_E or a good deal of X_E. Thus, the decision maker learns about X_E and the tradeoffs for problem (MOLP) that it implies. In this way, the decision maker is able, in theory, to make an informed decision when he or she chooses a most preferred solution from the generated set.

Unfortunately, there are several weaknesses in applying the vector maximization approach to problem (MOLP). This has led to only limited success with this approach in practice. These weaknesses are of at least four types.

First, the computational demands of finding X_E or representative subsets of X_E grow very rapidly with problem size. For example, the number of elements in $X_E \cap X_{ex}$ grows exponentially with problem size. As a result, applications of vector maximization algorithms have been limited to only relatively small problems [8,29,33,36].

Second, the sheer size of X_E or of representative subsets of X_E often precludes the possibility for the analyst of presenting these sets to the decision maker in a practical, meaningful way. While techniques for aiding with this presentation have been proposed [28], even with these techniques the challenge of communicating the generated set of efficient decisions to the decision maker can be daunting. Frequently, the decision maker can be confused or overwhelmed by the sheer size of the set (see, e.g., [7,35,36] and references therein).

Third, most vector maximization methods for problem (MOLP) use simplex method-like steps that call for burdensome bookkeeping or backtracking schemes. These schemes must keep an account of the points in X_E that have been generated, and often many of these generated points must be revisited in order to continue the generation process [7,19,36].

Fourth, all of the decision set-based vector maximization methods must deal with degenerate efficient extreme points. Usually, special burdensome procedures are used for this purpose. Even when all of X_E is generated, these procedures usually must be invoked, because, in order to find X_E, most vector maximization methods for problem (MOLP) first generate $X_E \cap X_{ex}$ [2,17,37,38].

Motivated by these difficulties, some researchers in recent years have proposed solving problem (MOLP) in the outcome space $\mathbb{R}^p$, rather than in the decision space $\mathbb{R}^n$. In particular, ideas and methods have recently been proposed for modifying the vector maximization approach for problem (MOLP) so as to generate all or parts of the efficient outcome set Y_E, rather than all or parts of the efficient decision set X_E, of the problem (see, e.g., [5,7,8,13–16]). The rationale for this approach is threefold.

First, the dimension p of the outcome space $\mathbb{R}^p$ is typically much smaller than the dimension n of the decision space $\mathbb{R}^n$, often by one or more orders of magnitude. As a result, Y_E is invariably much smaller than X_E and contains far fewer faces and extreme points than X_E [6,14–16]. Generating all or parts of Y_E can therefore be expected to be less demanding computationally than generating all or portions of X_E. In addition, presenting all or portions of Y_E to the decision maker can be expected to be easier and less overwhelming for the decision maker than presenting all or parts of X_E.

Second, empirical research has shown that, in practice, the decision maker generally prefers examining efficient outcomes rather than efficient decisions in order to search for a most preferred solution [7,8,15,16]. Many (if not most) methods that generate X_E also make Y_E available to the decision maker. Since Y_E is much simpler than X_E, however, methods that show only the outcome set Y_E to the decision maker are less burdensome for him or her.

Third, it has been shown that frequently C maps many points in X_E onto either a single outcome in Y_E or onto essentially equivalent outcomes in Y_E [3,6,14,15]. Thus, generating points directly from Y_E avoids risking redundant determinations of points in X_E.

To date, two implementable algorithms have been proposed for generating $Y_E \cap Y_{ex}$, where Y_{ex} denotes the set of all extreme points of the outcome set Y [7,8]. These algorithms have certain important advantages, and preliminary computational tests with one of them seem promising [8]. They both rely mainly upon an adaptation of a well-known outer approximation technique used in global optimization. However, global optimization research has shown that this outer approximation technique can be expected to require unacceptably large computation times and storage as p increases [22–24]. Thus it seems worthwhile to explore alternate approaches for generating $Y_E \cap Y_{ex}$.

In this article, we present and validate a basic weight set decomposition approach for generating $Y_E \cap Y_{ex}$. The idea of using weight set decomposition was first proposed by Gal and Nedoma [21] to deal with multiparametric linear programming. Later, Zeleny [39] proposed using this approach to generate $X_E \cap X_{ex}$. However, as explained, for instance, by Yu [37], Zeleny's approach can fail to generate $X_E \cap X_{ex}$. This is because Zeleny's weight set decomposition elements are not in one-to-one correspondence with the elements of $X_E \cap X_{ex}$.

Based upon a set of basic weight set decomposition steps, we develop a new algorithm, called the Weight Set Decomposition Algorithm, for generating $Y_E \cap Y_{ex}$. Since this algorithm generates the efficient extreme points of the outcome set $Y \subseteq \mathbb{R}^p$ rather than those of the decision set $X \subseteq \mathbb{R}^n$, it will potentially require less computational effort than methods for generating the set $X_E \cap X_{ex}$. Furthermore, the Weight Set Decomposition Algorithm is guaranteed to find all of $Y_E \cap Y_{ex}$ without necessitating the use of elaborate bookkeeping or backtracking schemes. These schemes are not needed because the elements of $Y_E \cap Y_{ex}$ are found by using global optimization search ideas rather than by repeated simplex method pivots from efficient extreme point to efficient extreme point.

This article is organized as follows. Preliminary results needed to present the steps of the basic weight set decomposition approach are given in Section 2. Section 3 presents and validates the basic weight set decomposition approach. In Section 4, based upon the weight set decomposition approach, the Weight Set Decomposition Algorithm for generating $Y_E \cap Y_{ex}$ is developed and presented. A sample problem is solved by this new algorithm in Section 5. The potential computational and practical benefits of the Weight Set Decomposition Algorithm and some conclusions are given in Section 6.

2. Preliminaries

Let $W^0 = \{w \in \mathbb{R}^p \mid w_j > 0,\ j = 1, 2, \ldots, p\}$. We refer to W^0 as the *weight set* associated with problem (MOLP). It is well known (see, e.g., [38]) that $x^0 \in X_E$ if and only if x^0 is an optimal solution for the linear program $\mathrm{LP}(w)$ for some

$w = w^0 \in W^0$ where, for any $w \in \mathbb{R}^p$, LP(w) is given by

LP(w)

$$\begin{array}{ll} \max & w^{\mathrm{T}} Cx, \\ \text{s.t.} & Ax \leqslant b, \\ & x \geqslant 0. \end{array}$$

For any $y \in Y$, define $W(y)$ by

$$W(y) = \{w \in \mathbb{R}^p \,|\, \langle w, y' \rangle \leqslant \langle w, y \rangle \text{ for all } y' \in Y\}.$$

From [30], for each $y \in Y$, $W(y)$ is the *normal cone* to Y at y. It is easy to show that x^0 is an optimal solution to problem LP(w^0) if and only if $w^0 \in W(y^0)$, where $y^0 = Cx^0$.

Suppose that $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}} = \{y^1, y^2, \ldots, y^q\}$, where, since Y is compact, we may assume that $q \geqslant 1$ [4]. From Theorem 3.1 in [9], we have the following result:

Theorem 1. $W^0 \subseteq \bigcup_{i=1}^{q} [W(y^i) \cap W^0]$.

Theorem 1 implies that

$$W^0 = \bigcup_{i=1}^{q} [W^0 \cap W(y^i)], \tag{1}$$

i.e., that the weight set W^0 can be decomposed into a union of subsets $[W^0 \cap W(y^i)]$, $i = 1, 2, \ldots, q$, of W^0. Furthermore, from Theorems 3.2 and 3.4 in [9], it follows that for each $i = 1, 2, \ldots, q$,

$$\operatorname{int}[W^0 \cap W(y^i)] \neq \emptyset$$

and that for each pair of distinct indices $i, j \in \{1, 2, \ldots, q\}$,

$$\operatorname{int}[W^0 \cap W(y^i)] \cap \operatorname{int}[W^0 \cap W(y^j)] = \emptyset,$$

where, for each $k \in \{1, 2, \ldots, q\}$, $\operatorname{int}[W^0 \cap W(y^k)]$ denotes the interior of $[W^0 \cap W(y^k)]$. As a result, in the decomposition (1) of W^0, the nonempty interiors of the sets $[W^0 \cap W(y^i)]$, $i = 1, 2, \ldots, q$, do not intersect. Furthermore, it follows that there is a one-to-one correspondence between the efficient extreme points y^i, $i = 1, 2, \ldots, q$, of the outcome set Y and the subsets $[W^0 \cap W(y^i)]$, $i = 1, 2, \ldots, q$, of the weight set W^0. We therefore refer to the decomposition (1) as the *outcome-based decomposition* (or *partition*) of the weight set W^0. The basic weight set decomposition steps to be presented in the following section are based upon the partition (1) of W^0.

To help show that the basic weight set decomposition approach to be presented in Section 2 is finite and valid, we will use the next result. The proof of this result follows from some results in [9] and, for ease of presentation, is not given here. We define a set T to be a *proper subset* of a set S when T is a nonempty subset of S and $T \neq S$.

Theorem 2. *Let I be a proper subset of $\{1, 2, \ldots, q\}$. Then*

$$\left[W^0 \cap \left(\bigcup_{i \in I} W(y^i)\right)\right]$$

is a proper subset of W^0.

The basic weight set decomposition approach is iterative and involves finding a new weight vector $w^k \in W^0$ at each iteration $k = 1, 2, \ldots, q$. For each $k \in \{1, 2, \ldots, q\}$, after w^k is found, the corresponding extreme point $y^k \in Y_{\mathrm{E}}$ is found, and $W^k \cap W(y^k)$ is removed from W^k, where W^k represents the subset of W^0 that remained at the beginning of iteration k. Theorem 2 guarantees that after k iterations, if not all of the points in $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ have been found, then $W^{k+1} \neq \emptyset$.

3. Basic weight set decomposition approach

The steps of the basic weight set decomposition approach are based upon the partition (1) of W^0. At each iteration, the approach first either finds a weight vector in the weight set W^0 that leads to some unexplored efficient extreme point of the outcome set, or it detects that all points in $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ have been found.

Two questions arise. One is how to find a weight vector in W^0 that will lead to an unexplored point in $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ or to determine that no such weight vector exists. We will address this question later. The other question is how to find a point in $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$, given a weight vector in W^0. The following two theorems help answer the latter question:

Theorem 3. *Assume that w belongs to W^0. If x^* is the unique optimal solution to problem* LP(w), *then $y^* = Cx^*$ belongs to $Y_E \cap Y_{ex}$.*

Proof. Since x^* is the unique optimal solution to problem LP(w), $y^* = Cx^*$ is the unique optimal solution to the problem

$$\begin{array}{ll} \max & \langle w, y \rangle \\ \text{s.t.} & y \in Y. \end{array}$$

From linear programming theory, this implies that $y^* \in Y_{ex}$. Furthermore, from [38], since $w \in W^0$, this also implies that $y^* \in Y_E$. $\square$

Theorem 4. *Assume that w belongs to W^0 and that x^* is an optimal solution to problem* LP(w). *If y' is an extreme point of the set*

$$Y \cap \{y \in \mathbb{R}^p \,|\, \langle w, y \rangle = w^{\mathrm{T}} C x^*\},$$

then $y' \in Y_E \cap Y_{ex}$.

Proof. The proof that $y' \in Y_E$ follows by using the same argument used in the proof of Theorem 3. To show that $y' \in Y_{ex}$, we use a proof by contradiction. We therefore suppose that $y' \notin Y_{ex}$. By definition, this implies that we may choose distinct points $y^1, y^2 \in Y$ such that

$$y' = ty^1 + (1-t)y^2$$

for some scalar t such that $0 < t < 1$. Choose such a scalar t. Then

$$\langle w, y' \rangle = t\langle w, y^1 \rangle + (1-t)\langle w, y^2 \rangle. \tag{2}$$

We claim that $\langle w, y^1 \rangle = \langle w, y^2 \rangle$ must be true. To show this claim, assume, to the contrary, that $\langle w, y^1 \rangle \neq \langle w, y^2 \rangle$. Then, without loss of generality, we may assume that $\langle w, y^1 \rangle < \langle w, y^2 \rangle$. From (2), since $t > 0$, this implies that

$$\langle w, y' \rangle < t\langle w, y^2 \rangle + (1-t)\langle w, y^2 \rangle = \langle w, y^2 \rangle. \tag{3}$$

Since $y' \in Y \cap \{y \in \mathbb{R}^p \,|\, \langle w, y \rangle = w^{\mathrm{T}} C x^*\}$, $\langle w, y' \rangle = w^{\mathrm{T}} C x^*$. Because $y^2 \in Y$, we may choose a vector $x^2 \in X$ such that $y^2 = Cx^2$. From (3), the latter two sentences imply that

$$w^{\mathrm{T}} C x^* < w^{\mathrm{T}} C x^2.$$

However, since $x^2 \in X$, this contradicts the assumption that x^* is an optimal solution to problem LP(w), so that the claim is established.

From (2), since $\langle w, y^1 \rangle = \langle w, y^2 \rangle$, it follows that

$$\langle w, y' \rangle = \langle w, y^1 \rangle = \langle w, y^2 \rangle. \tag{4}$$

Also, since $y' \in Y \cap \{y \in \mathbb{R}^p \,|\, \langle w, y \rangle = w^{\mathrm{T}} C x^*\}$, we know that $\langle w, y' \rangle = w^{\mathrm{T}} C x^*$. Together with (4), since $y^1, y^2 \in Y$, this implies that

$$y^1, y^2 \in Y \cap \{y \in \mathbb{R}^p \,|\, \langle w, y \rangle = w^{\mathrm{T}} C x^*\}. \tag{5}$$

From (5), since $y' = ty^1 + (1-t)y^2$, where $0 < t < 1$, and $y^1 \neq y^2$, y' is not an extreme point of $Y \cap \{y \in \mathbb{R}^p \,|\, \langle w, y \rangle = w^{\mathrm{T}} C x^*\}$. Since this contradicts the assumption of the theorem that y' is an extreme point of $Y \cap \{y \in \mathbb{R}^p \,|\, \langle w, y \rangle = w^{\mathrm{T}} C x^*\}$, the theorem is proven. $\square$

The steps of the basic weight set decomposition approach for generating $Y_E \cap Y_{ex}$ are as follows:

- *Step 1*. Set $W^1 = W^0$. Choose any point $w^1 \in W^1$. Find any optimal extreme point solution $\bar{x}^1$ to the linear program LP(w^1). Set $k := 1$, $\mathrm{EX}^0 := \emptyset$, $\mathrm{EY}^0 := \emptyset$.
- *Step 2*. If $\bar{x}^k$ is the unique optimal solution to the linear program LP(w^k), set $x^k = \bar{x}^k$ and $y^k = C\bar{x}^k$. Otherwise, find any extreme point y^k of the polyhedron $Y \cap \{y \in \mathbb{R}^p \,|\, \langle w^k, y \rangle = (w^k)^{\mathrm{T}} C\bar{x}^k\}$, and any extreme point x^k of X such that $y^k = Cx^k$. Set $\mathrm{EX}^k = \mathrm{EX}^{k-1} \cup \{x^k\}$, $\mathrm{EY}^k = \mathrm{EY}^{k-1} \cup \{y^k\}$ and $W^{k+1} = W^k \setminus W(y^k)$.
- *Step 3*. If $W^{k+1} \neq \emptyset$, find any point w^{k+1} in W^{k+1} and go to Step 4. Otherwise, stop: $Y_E \cap Y_{ex} = \mathrm{EY}^k$.
- *Step 4*. Find any optimal extreme point solution $\bar{x}^{k+1}$ to the linear program LP(w^{k+1}). Set $k := k + 1$ and go to Step 2.

Notice that the feasible region of problem LP(w) is a nonempty, compact polyhedron. Thus, from linear programming theory, for any point w in $\mathbb{R}^p$, problem LP(w) has at least one optimal extreme point solution. Hence, Steps 1 and 4 of the basic weight set decomposition approach are well defined.

In Step 2, if $\bar{x}^k$ is the unique optimal solution to problem LP(w^k), then, from Theorem 3, $y^k = C\bar{x}^k$ belongs to $Y_E \cap Y_{ex}$. Otherwise, Step 2 calls for finding an extreme point y^k of the nonempty, compact polyhedron $Y \cap \{y \in \mathbb{R}^p \,|\, \langle w^k, y\rangle = (w^k)^{\mathrm{T}} C\bar{x}^k\}$ and an extreme point x^k of X such that $y^k = Cx^k$. From linear programming theory and from Benson [6], since X is nonempty and compact, such extreme points x^k and y^k must exist. Therefore, Step 2 of the basic weight set decomposition approach is well defined. Furthermore, when $\bar{x}^k$ is not the unique optimal solution to problem LP(w^k), by Theorem 4, the extreme point y^k found in Step 2 belongs to $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$.

The basic weight set decomposition approach is finite and valid by the following result:

Theorem 5. *The basic weight set decomposition approach terminates at some iteration $q \in \{1, 2, \ldots\}$ and, upon termination, $EY^q = Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$, where q is the number of points in $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$.*

Proof. See Appendix A. □

Most of the steps of the basic weight set decomposition approach are relatively simple to implement. In Step 3, however, for a fixed k, the need to find, if it exists, a point w^{k+1} in $W^{k+1} = W^k \setminus W(y^k)$ is relatively challenging. This is largely due to the fact that for each $k \geqslant 1$, when $W^k \setminus W(y^k)$ is nonempty, it is a nonclosed, nonconvex cone.

Therefore, in the remainder of this section, we will show how one can implement Steps 1, 2 and 4 of the approach. Subsequently, we will address the implementation of Step 3.

In Step 1 or Step 4, an extreme point optimal solution $\bar{x}^k$ must be found to linear program LP(w^k), where $k \geqslant 1$. This can be easily accomplished, for instance, by the simplex method.

In Step 2 of the approach, for $k \geqslant 1$, it is sometimes necessary to find an extreme point y^k of the polyhedron $Y \cap \{y \in \mathbb{R}^p \,|\, \langle w^k, y\rangle = (w^k)^{\mathrm{T}} C\bar{x}^k\}$ and to find an extreme point x^k of X such that $y^k = Cx^k$. To find these points, we may apply a linear programming-based procedure of Benson and Sun [10].

To explain how the procedure in [10] can be used to find the points y^k and x^k of the previous paragraph, let

$$D = [\,C \quad 0_{p\times m}\,],$$

$$\bar{A} = \begin{bmatrix} A & E_{m\times m} \\ (w^k)^{\mathrm{T}} C & 0_{1\times m} \end{bmatrix},$$

$$\bar{b} = \begin{bmatrix} b \\ (w^k)^{\mathrm{T}} C\bar{x}^k \end{bmatrix},$$

and

$$Z = \{z \in \mathbb{R}^{n+m} \,|\, \bar{A}z = \bar{b},\ z \geqslant 0\},$$

where $0_{p\times m}$, $0_{1\times m}$ and $E_{m\times m}$ denote the $p \times m$ matrix of zeroes, the $1 \times m$ vector of zeroes, and the $m \times m$ identity matrix, respectively, and $z^{\mathrm{T}} = (x^{\mathrm{T}}, s^{\mathrm{T}})$, where $x \in \mathbb{R}^n$ and $s \in \mathbb{R}^m$ are the variable vectors. Notice that since $Y = \{Cx \,|\, x \in X\}$, y is an element of the polyhedron $Y \cap \{y \in \mathbb{R}^p \,|\, \langle w^k, y\rangle = (w^k)^{\mathrm{T}} C\bar{x}^k\}$ if and only if $y = Dz$ for some $z = (x^{\mathrm{T}}, s^{\mathrm{T}})^{\mathrm{T}}$ such that $\bar{A}z = \bar{b}$ and $z \geqslant 0$, i.e., if and only if y is an element of the polyhedron $\{Dz \,|\, z \in Z\}$. This implies that y^k is an extreme point of the polyhedron $Y \cap \{y \in \mathbb{R}^p \,|\, \langle w^k, y\rangle = (w^k)^{\mathrm{T}} C\bar{x}^k\}$ if and only if y^k is an extreme point of the polyhedron

$$DZ \equiv \{Dz \,|\, z \in Z\}. \tag{6}$$

The linear programming procedure in [10] is designed to find extreme points of polyhedra of the form (6). When applied to (6), this procedure is as follows:

- *Step 1*. Find any optimal extreme point solution $(z^1)^{\mathrm{T}} = [(x^1)^{\mathrm{T}}, (s^1)^{\mathrm{T}}]$ to the linear program

  LPD(1)

  $$\begin{array}{ll} \min & \langle D_1, z\rangle \\ \text{s.t.} & z \in Z, \end{array}$$

  where D_1 denotes row 1 of the matrix D. If z^1 is the unique optimal solution to linear program LPD(1), then stop: $y^k = Cx^1$ is an extreme point of $Y \cap \{y \in \mathbb{R}^p \,|\, \langle w^k, y\rangle = (w^k)^{\mathrm{T}} C\bar{x}^k\}$. Otherwise, let θ_1 denote the optimal value of problem LPD(1), set $i = 2$, and go to Step i.
- *Step i ($i \geqslant 2$)*. Find any optimal extreme point solution $(z^i)^{\mathrm{T}} = [(x^i)^{\mathrm{T}}, (s^i)^{\mathrm{T}}]$ and the optimal value θ_i of the linear program

  LPD(i)

$$\begin{aligned} \min \quad & \langle D_i, z\rangle \\ \text{s.t.} \quad & \langle D_t, z\rangle = \theta_t, \\ & \quad t = 1,2,\ldots,i-1, \\ & z \in Z, \end{aligned}$$

where D_t denotes row t of D, $t = 1,2,\ldots,i$. If z^i is the unique optimal solution to linear program LPD(i) or if $i = p$, then stop: $y^k = Cx^i$ is an extreme point of $Y \cap \{y \in \mathbb{R}^p \,|\, \langle w^k, y\rangle = (w^k)^{\mathrm{T}} C\bar{x}^k\}$. Otherwise, set $i = i + 1$ and go to Step i.

The validity of this procedure follows from [10]. In particular, from [10], the procedure is finite and, when it stops in Step i, $1 \leqslant i \leqslant p$, $y^k = Cx^i$ is an extreme point of $Y \cap \{y \in \mathbb{R}^p \,|\, \langle w^k, y\rangle = (w^k)^{\mathrm{T}} C\bar{x}^k\}$. Notice that the procedure finds this extreme point after solving at most p linear programming problems. Furthermore, by the following result, it also finds an extreme point x^k of X such that $y^k = Cx^k$.

Theorem 6. *Let $x^k = x^i$, where i is the step number in which the procedure above terminates. Then x^k is an extreme point of X, and $y^k = Cx^k$.*

Proof. The fact that $y^k = Cx^k$ follows from the statement of the procedure and the definition of x^k. By definition of Z and the statement of the procedure, it is clear that $x^k = x^i \in X$.

To show that $x^k = x^i$ is an extreme point of X, we use a proof by contradiction. Towards this end, suppose, to the contrary, that $x^k = x^i$ is not an extreme point of X. Then we may choose $\alpha \in \mathbb{R}$ such that $0 < \alpha < 1$ and distinct points $x', x'' \in X$ such that

$$x^k = \alpha x' + (1-\alpha)x''. \tag{7}$$

Since $(z^i)^{\mathrm{T}} = [(x^i)^{\mathrm{T}}, (s^i)^{\mathrm{T}}] \in Z$ and $x^k = x^i$, it follows that

$$(w^k)^{\mathrm{T}} Cx^k = (w^k)^{\mathrm{T}} C\bar{x}^k. \tag{8}$$

Recall that the given point $\bar{x}^k$ in the procedure is an optimal solution to problem LP(w^k). Since $x^k \in X$, together with (8) this implies that x^k is also an optimal solution to problem LP(w^k). As a result, since x' and x'' are in X and $0 < \alpha < 1$, using (7) it is easy to see that both x' and x'' are also optimal solutions to problem LP(w^k). By the definition of X, this implies that x' and x'' are distinct points in the set

$$\{x \in \mathbb{R}^n \,|\, Ax \leqslant b,\ x \geqslant 0,\ (w^k)^{\mathrm{T}} Cx = (w^k)^{\mathrm{T}} C\bar{x}^k\}. \tag{9}$$

Set $(z')^{\mathrm{T}} = [(x')^{\mathrm{T}}, (s')^{\mathrm{T}}]$ and $(z'')^{\mathrm{T}} = [(x'')^{\mathrm{T}}, (s'')^{\mathrm{T}}]$, where $s' = b - Ax'$ and $s'' = b - Ax''$. Then, by the definition of D, since x' and x'' are distinct points in the set described by (9), z' and z'' are distinct points in Z. In addition, from (7) and the definitions of z' and z'', since $x^k = x^i$ and $(z^i)^{\mathrm{T}} = [(x^i)^{\mathrm{T}}, (s^i)^{\mathrm{T}}]$, it follows that

$$z^i = \alpha z' + (1-\alpha)z''. \tag{10}$$

Therefore,

$$Dz^i = \alpha Dz' + (1-\alpha)Dz''.$$

Since z^i is an optimal solution to linear program LPD(i), this implies that for each $t = 1,2,\ldots,i$,

$$\theta_t = \langle D_t, z^i\rangle = \alpha\langle D_t, z'\rangle + (1-\alpha)\langle D_t, z''\rangle. \tag{11}$$

Since z' and z'' are distinct points in Z and $0 < \alpha < 1$, if we set $t = 1$ in (11), we see that z' and z'' are distinct optimal solutions to problem LPD(1). From this, by setting $t = 2$ in (11), it follows that since z' and z'' are distinct points in Z and $0 < \alpha < 1$, z' and z'' are distinct optimal solutions to problem LPD(2). This, is turn, implies, by setting $t = 3$ in (11), since z' and z'' are distinct points in Z and $0 < \alpha < 1$, that z' and z'' are distinct optimal solutions to problem LPD(3). By continuing in this fashion, we see that z' and z'' are distinct optimal solutions to problem LPD(i). From (10), since $0 < \alpha < 1$ and z^i is an optimal solution to problem LPD(i), this implies that z^i is not an extreme point of the feasible region of problem LPD(i). From Step i of the procedure, however, z^i is an extreme point of the feasible region of problem LPD(i). We thus have a contradiction.

Therefore, $x^k = x^i$ must be an extreme point of X, and the proof is complete. □

Implementing Step 3 of the basic weight set decomposition approach is the most challenging implementation task of the approach. This is because for each $k \geqslant 1$, when $W^{k+1} = W^k \setminus W(y^k)$ is nonempty, it is a nonclosed, nonconvex cone. We will present a method for implementing Step 3 of the approach in the following section. This method will yield the Weight Set Decomposition Algorithm for generating $Y_E \cap Y_{ex}$.

4. The Weight Set Decomposition Algorithm

The Weight Set Decomposition Algorithm for generating $Y_E \cap Y_{ex}$ implements Step 3 of the basic weight set decomposition approach by using a new global tree search technique that we have developed. Recall that q represents the number of elements in $Y_E \cap Y_{ex}$. To develop the tree search technique, we need to present conditions under which $w \notin W(y^i)$, where $i \in \{1, 2, \ldots, q\}$. This is because for any $k \geqslant 1$, to implement Step 3 of the basic weight set decomposition approach, we need to identify

$$W^{k+1} = W^0 \Big\backslash \bigcup_{i=1}^{k} W(y^i).$$

The next three results help provide a practical necessary and sufficient condition for a point $w \in \mathbb{R}^p$ to satisfy $w \notin W(y)$, where $y \in Y$. These results all assume that $y \in Y$ and that a point $x \in X$ is available such that $y = Cx$.

To help present and prove these three results, we need some additional notation. Suppose that $x \in X$, $y \in Y$, and $y = Cx$. For each $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$, let A_i and E_j denote row i of A and row j of E, respectively, where E represents the $n \times n$ identity matrix. Let

$$\mathrm{ID}(x) = \{i \in \{1, 2, \ldots, m\} \,|\, A_i x = b_i\},$$

and let

$$I^0(x) = \{j \in \{1, 2, \ldots, n\} \,|\, x_j = 0\}.$$

If $\mathrm{ID}(x) \neq \emptyset$, let A^I denote the matrix whose rows are A_i, $i \in \mathrm{ID}(x)$. If $\mathrm{ID}(x) = \emptyset$, let A^I equal the scalar 0. Similarly, let E^I represent the matrix whose rows are E_j, $j \in I^0(x)$, where E^I equals the scalar 0 if $I^0(x) = \emptyset$. Finally, let

$$D = \{d \in \mathbb{R}^n \,|\, A^I d \leqslant 0,\ E^I d \geqslant 0\}. \tag{12}$$

Lemma 1. *Assume that $w \in \mathbb{R}^p$. Then $w \notin W(y)$ if and only if there exists a vector $d \in D$ such that $w^{\mathrm{T}} C d > 0$.*

Proof. See Appendix A. □

For any set T, let

$$\mathrm{cone}(T) = \{\alpha t \,|\, \alpha \geqslant 0,\ t \in T\}.$$

The following result provides a means for generating the set D defined by (12).

Lemma 2. *Let D be defined by* (12). *Then*

$$D = \mathrm{cone}(X - \{x\}).$$

Proof. See Appendix A. □

Notice that in Lemmas 1 and 2, it is assumed that $x \in X$ and $y = Cx$. Assume now that, in addition, x is an extreme point of X. Let $S_{at}(x)$ denote the set of all extreme points of X adjacent to x. Then, using Lemmas 1 and 2, we obtain the following practical necessary and sufficient condition for a weight vector $w \in \mathbb{R}^p$ to satisfy $w \notin W(y)$.

Theorem 7. *A point $w \in \mathbb{R}^p$ satisfies $w \notin W(y)$ if and only if there exists a point $\bar{x} \in S_{at}(x) - \{x\}$ such that $w^{\mathrm{T}} C \bar{x} > 0$.*

Proof. From Lemma 1 $w \notin W(y)$ if and only if there exists a vector $d \in D$ such that $w^{\mathrm{T}} C d > 0$. By Lemma 2, this implies that $w \notin W(y)$ if and only if $w^{\mathrm{T}} C d > 0$ for some $d \in \mathrm{cone}(\{X - \{x\})$. Let $S_{at}(x) - \{x\} = \{x^1 - x, x^2 - x, \ldots, x^r - x\}$. Then, since x is an extreme point of X and X is compact,

$$\begin{aligned}\mathrm{cone}(X - \{x\}) &= \{z \in \mathbb{R}^n \,|\, z = \sum_{v=1}^{r} \alpha_v (x^v - x) \\ &\qquad \text{for some } \alpha_v \geqslant 0, v = 1, 2, \ldots, r\}.\end{aligned} \tag{13}$$

Suppose that there exists a point $d \in \mathrm{cone}(X - \{x\})$ such that $w^{\mathrm{T}}Cd > 0$. Then, by (13), there exist $\alpha_v \geqslant 0,\ v = 1,2,\ldots,r$, such that

$$w^{\mathrm{T}}C\left[\sum_{v=1}^{r}\alpha_v(x^v - x)\right] > 0.$$

This implies that

$$\sum_{v=1}^{r}\alpha_v w^{\mathrm{T}}C(x^v - x) > 0.$$

Since $\alpha_v \geqslant 0,\ v = 1,2,\ldots,r$, this implies that $w^{\mathrm{T}}C(x^v - x) > 0$ for at least one $v \in \{1,2,\ldots,r\}$. By definition of $S_{\mathrm{at}}(x) - \{x\}$, this means that $w^{\mathrm{T}}C\bar{x} > 0$ for some $\bar{x} \in S_{\mathrm{at}}(x) - \{x\}$.

Suppose now that $w^{\mathrm{T}}C\bar{x} > 0$ for some $\bar{x} \in S_{\mathrm{at}}(x) - \{x\}$. Then, for some $v \in \{1,2,\ldots,r\}$, $\bar{x} = x^v - x$. From (13), this implies that $\bar{x} \in \mathrm{cone}(X - \{x\})$. Hence, $w^{\mathrm{T}}C\bar{x} > 0$, where $\bar{x} \in \mathrm{cone}(X - \{x\})$.

The previous two paragraphs imply that $w^{\mathrm{T}}Cd > 0$ for some $d \in \mathrm{cone}(X - \{x\})$ if and only if $w^{\mathrm{T}}C\bar{x} > 0$ for some $\bar{x} \in S_{\mathrm{at}}(x) - \{x\}$. From the first paragraph of the proof, this implies that $w \notin W(y)$ if and only if $w^{\mathrm{T}}C\bar{x} > 0$ for some $\bar{x} \in S_{\mathrm{at}}(x) - \{x\}$ so that the proof is complete. □

Remark 1. Notice from Theorem 7 that if $y \in Y$ and x is an extreme point of X such that $y = Cx$, then w is *not* in $W(y)$ if and only if $w^{\mathrm{T}}C\bar{x} > 0$ for some $\bar{x} = x^v - x$, where x^v is a neighboring extreme point to x in X. Thus, under these conditions, we can check whether or not $w \notin W(y)$ by generating the neighbors $\{x^v \mid v = 1,2,\ldots,r\}$ to x in X and examining the numbers $w^{\mathrm{T}}C(x^v - x),\ v = 1,2,\ldots,r$, to see if any of them exceeds 0. If so, $w \notin W(y)$. Otherwise, $w \in W(y)$. When x is a nondegenerate extreme point of X, $S_{\mathrm{at}}(x) - \{x\}$ can be easily found by determining the simplex tableau for x. If x is degenerate, this tableau and some special steps can be invoked to find $S_{\mathrm{at}}(x) - \{x\}$ (see [20] and references therein).

Remark 2. Suppose that $k \in \{1,2,\ldots,q\}$, where, as before, q denotes the number of points in $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$. Theorem 7 provides the theoretical basis for generating a point $w^{k+1} \in W^{k+1}$ in Step 3 of the basic weight set decomposition approach, or showing that $W^{k+1} = \emptyset$. To see this, notice that upon entering Step 3, for each $i = 1,2,\ldots,k$, the approach has generated extreme points $y^i \in Y$ and $x^i \in X$ such that $y^i = Cx^i$. Notice also that at this point of the approach,

$$W^{k+1} = W^0 \Big\backslash \bigcup_{i=1}^{k} W(y^i).$$

Therefore, from Theorem 7, $w \in W^{k+1}$ if and only if $w > 0$ and there exist points $\bar{x}^i \in S_{\mathrm{at}}(x^i) - \{x^i\}$, $i = 1,2,\ldots,k$, such that $w^{\mathrm{T}}C\bar{x}^i > 0$ for all $i = 1,2,\ldots,k$. Otherwise, $W^{k+1} = \emptyset$.

Remark 3. Assume that $k \in \{2,3,\ldots,q\}$, and that h is an integer that satisfies $1 \leqslant h < k$. Suppose that there is no $w > 0$ and no set of points $\{\bar{x}^1, \bar{x}^2,\ldots,\bar{x}^h\}$ such that $\bar{x}^i \in S_{\mathrm{at}}(x^i) - \{x^i\},\ i = 1,2,\ldots,h$, and

$$w^{\mathrm{T}}C\bar{x}^i > 0 \tag{14}$$

for each $i = 1,2,\ldots,h$. Then it follows that there is no $w > 0$ and no set of points $\{\bar{x}^1,\bar{x}^2,\ldots,\bar{x}^h,\ldots,\bar{x}^k\}$ such that $\bar{x}^i \in S_{\mathrm{at}}(x^i) - \{x^i\},\ i = 1,2,\ldots,h,\ldots,k$, and such that (14) holds for each $i = 1,2,\ldots,h,\ldots,k$.

Suppose that $k \in \{1,2,\ldots,q\}$. Using Remarks 1–3, we have developed a special global tree search method for finding a point $w \in W^{k+1}$ or showing that $W^{k+1} = \emptyset$. Notice that for any fixed $\bar{x} \in \mathbb{R}^n$, $w^{\mathrm{T}}C\bar{x} > 0$ holds for some $w > 0$ if and only if the linear inequality system

$$w^{\mathrm{T}}C\bar{x} \geqslant 1,$$
$$w \geqslant e$$

is feasible, where $e \in \mathbb{R}^p$ denotes the vector of ones. The main process used in the tree search is to seek a set of points $\{\hat{x}^1,\hat{x}^2,\ldots,\hat{x}^k\}$ such that $\hat{x}^i \in S_{\mathrm{at}}(x^i) - \{x^i\},\ i = 1,2,\ldots,k$, and for which the linear system

$$w^{\mathrm{T}}C\hat{x}^i \geqslant 1, \quad i = 1,2,\ldots,k, \tag{15}$$
$$w \geqslant e \tag{16}$$

has at least one solution w. If a set of such points $\hat{x}^i \in S_{\rm at}(x^i) - \{x^i\}$, $i = 1, 2, \ldots, k$, exists, then the tree search method will find such a set, and, for this set, it finds a solution w for (15) and (16). From Remark 2, $w \in W^{k+1}$ will hold. If no such set of points is found, then, by the same remark, $W^{k+1} = \emptyset$. Furthermore, by using Remark 3, the tree search for such a set of points can take advantage of searches that were performed earlier for values h of k satisfying $1 \leqslant h < k$.

The tree search method will be integrated into the basic weight set decomposition approach to yield the Weight Set Decomposition Algorithm. In each iteration $k \in \{1, 2, \ldots q\}$, after the extreme point x^k in Step 2 of the basic approach is found, the algorithm also finds the set $D'(k) = S_{\rm at}(x^k) - \{x^k\}$ by using Remark 1. Thus, by iteration k, the algorithm has found the sets

$$D'(i) = S_{\rm at}(x^i) - \{x^i\}, \quad i = 1, 2, \ldots, k.$$

Assume that

$$S_{\rm at}(x^i) - \{x^i\} = \{x^{i,1} - x^i, x^{i,2} - x^i, \ldots, x^{i,t_i} - x^i\},$$
$$i = 1, 2, \ldots, k.$$

The sets

$$D'(i) = \{x^{i,1} - x^i, x^{i,2} - x^i, \ldots, x^{i,t_i} - x^i\},$$
$$i = 1, 2, \ldots, k,$$

form the basis of a tree, a schematic of which is shown in Fig. 1.

To explain the tree search, let $k \in \{1, 2, \ldots, q\}$. Notice in Fig. 1 that the tree has a root node. In addition, after k iterations, the tree has k levels. During the first iteration, for each of the t_1 elements of $D'(1) = S_{\rm at}(x^1) - \{x^1\} = \{x^{1,1} - x^1, x^{1,2} - x^1, \ldots, x^{1,t_1} - x^1\}$, a branch is created from the root node to a new node corresponding to an element of $D'(1)$. The nodes corresponding to $D'(1)$ are said to be at *level 1* of the tree. During the second iteration, for each of the nodes corresponding to the elements of $D'(1)$ that have not been eliminated (the elimination process is explained below), t_2 branches from the node to the nodes corresponding to the elements of $D'(2)$ are created. The nodes corresponding to $D'(2)$ are said to be at *level 2* of the tree. Notice that there are up to $t_1 t_2$ paths p from the root node to a node at level 2 of the tree. In general, at iteration i, $1 \leqslant i \leqslant k$, for each remaining node corresponding to an element of $D'(i-1)$, t_i branches from the node to the nodes corresponding to the elements of $D'(i)$ are created, where $D'(0)$ is the set containing simply the root node. The nodes corresponding to $D'(i)$ are said to be at *level i* of the tree. In this way, at iteration i, $1 \leqslant i \leqslant k$, up to $t_1 t_2 \cdots t_i$ paths p from the root node to level i have been created, each of the form

$$p = (\hat{x}^1, \hat{x}^2, \ldots, \hat{x}^i), \tag{17}$$

where, for each $s = 1, 2, \ldots, i$,

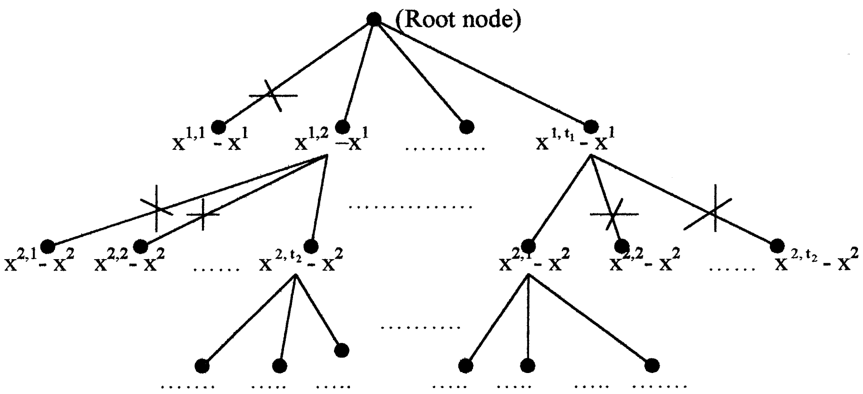

Fig. 1. Schematic of tree.

$\hat{x}^s \in \{x^{s,1} - x^s, x^{s,2} - x^s, \ldots, x^{s,t_s} - x^s\}$.

Thus, each element $\hat{x}^s$, $s = 1, 2, \ldots, i$, of a path from the root node to level i is of the form $x^{s,j} - x^s$, where x^s is an extreme point of X found by the algorithm in iteration s, and $x^{s,j}$ is an extreme point of X adjacent to x^s which, by using Remark 1, is also found by the algorithm. For each $i \in \{1, 2, \ldots, k\}$, let L_i denote the set of all remaining paths p of the form (17) from the root node to level i. Notice that $L_1 \subseteq D'(1)$ and, for each $k \in \{1, 2, \ldots, q\}$, $L_k \subseteq \{(p, p') \,|\, p \in L_{k-1}$ and $p' \in D'(k)\}$.

During the first iteration of the algorithm, we initially set $L_1 = D'(1)$. Next, a search of the paths $p^j = x^{1,j} - x^1$, $j = 1, 2, \ldots, M$, where $M = t_1$, is performed; i.e., in (15) and (16), with $i = 1$, we set $\hat{x}^1$ equal to p^j, $j = 1, 2, \ldots, M$, one of a time, and for each j, we determine whether or not (15) and (16) has a feasible solution w. If not, the branch from the root node to the corresponding level 1 node $x^{1,j} - x^1$ is eliminated, and, to signify this, L_1 is set equal to $L_1 \setminus \{p^j\}$. This is because in subsequent iterations $k = 2, 3, \ldots, q$, by Remark 3, no paths $p = (p^j, \hat{x}_2, \ldots, \hat{x}_k)$ can yield a system (15) and (16) that has a solution w. On the other hand, as soon as a path p^j, $j \in \{1, 2, \ldots, M\}$ is found such that (15) and (16) has a feasible solution w, we find such a solution w^* and we set $w^2 = w^*$. This is because in this case, by Remark 2, $w^* \in W^2$ and the algorithm continues.

In iteration k of the algorithm, we initially set $L_k = \{(p, p') \,|\, p \in L_{k-1}$ and $p' \in D'(k)\}$. Next, a search of the paths $p^j = (\hat{x}^1, \hat{x}^2, \ldots, \hat{x}^k)$, $j = 1, 2, \ldots M$, is undertaken, where $\hat{x}^i \in S_{\mathrm{at}}(x^i) - \{x^i\}$, $i = 1, 2, \ldots, k$, and M represents the number of elements in L_k. This search involves, for each path p^j, determining whether or not (15) and (16) has a feasible solution w. If not, L_k is set equal to $L_k \setminus p^j$ for reasons similar to those explained above for iteration 1. If so, then, by Remark 2, we set $w^{k+1} = w^*$, where w^* is a feasible solution to (15) and (16), and we are guaranteed that $w^{k+1} \in W^{k+1}$.

If the search of the paths p^j, $j = 1, 2, \ldots, M$, in L_k reveals that no path $p^j = (\hat{x}^1, \hat{x}^2, \ldots, \hat{x}^k)$ yields a feasible system (15) and (16), then, by Remark 2, $W^{k+1} = \emptyset$. In this case, the algorithm terminates, because, by Theorem 5, no more elements of $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ remain to be found.

By integrating the tree search method into the basic weight set decomposition approach, we obtain the following algorithm for generating $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$.

Weight Set Decomposition Algorithm

- *Step 1*. Set $W^1 = W^0$. Choose any point $w^1 \in W^1$. Find any optimal extreme point solution $\bar{x}^1$ to the linear programming problem LP(w^1). Set $k = 1$, $\mathrm{EX}^0 = \emptyset$ and $\mathrm{EY}^0 = \emptyset$.
- *Step 2*. If $\bar{x}^k$ is the unique optimal solution to problem LP(w^k), set $x^k = \bar{x}^k$ and $y^k = C\bar{x}^k$. Otherwise, find x^k and y^k by using the linear programming-based procedure given in Section 3. Find $S_{\mathrm{at}}(x^k)$, and let $D'(k) = S_{\mathrm{at}}(x^k) - \{x^k\}$. Set $\mathrm{EX}^k = \mathrm{EX}^{k-1} \cup \{x^k\}$ and $\mathrm{EY}^k = \mathrm{EY}^{k-1} \cup \{y^k\}$.
- *Step 3*.
  - *3.1*. If $k = 1$, set $L_k = D'(1)$. Otherwise set $L_k = \{(p, p') \,|\, p \in L_{k-1}$ and $p' \in D'(k)\}$. Let M denote the number of elements in L_k. Set $j = 1$.
  - *3.2*. Choose $p^j \in L_k$. Suppose that $p^j = (\hat{x}^1, \hat{x}^2, \ldots, \hat{x}^k)$.
  - *3.3*. Determine whether or not the linear system (15) and (16) has a feasible solution w^*. If so, set $w^{k+1} = w^*$ and go to Step 4. If not, set $L_k = L_k \setminus \{p^j\}$. If $j < M$, set $j = j + 1$ and go to Step 3.2. If $j = M$, stop: $\mathrm{EY}^k = Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$.
- *Step 4*. Find any optimal extreme point solution $\bar{x}^{k+1}$ to the linear programming problem LP(w^{k+1}). Set $k = k + 1$ and go to Step 2.

From our development and discussions in this section and the previous section, the Weight Set Decomposition Algorithm if finite and valid.

5. Computational considerations and sample problem

Notice that to execute the algorithm, one needs mainly to solve linear programming problems and to solve linear systems of the form (15) and (16), or to show that systems such as this have no solution. Let us examine briefly the computational demands imposed by these requirements.

For each efficient extreme point y^k in Y found by the algorithm, the linear program $\mathrm{LP}(w^k)$ must be solved. In some cases, up to p additional linear programs of the form $\mathrm{LPD}(i)$, $i \in \{1, 2, \dots, p\}$, must also be solved to generate y^k. It is known that the number of points in $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ grows exponentially with problem size. In theory, then, the number of linear programs that must be solved by the algorithm can also grow exponentially, especially as the dimension p of Y increases. However, since p is generally much smaller than n, it is expected that the computational demands of this growth will not increase as rapidly as they might if the method were decision set-based. Notice also that, if the simplex method is used, after solving $\mathrm{LP}(w^1)$, subsequent solutions of $\mathrm{LP}(w^k)$, $k = 2, 3, \dots, q$, can more easily be solved by choosing an optimal basis of the previous problem as the initial basis for the simplex method solution of the current problem.

To find each new weight w^{k+1} or to show that no such weight exists, the tree search method of Step 3 must be executed. At any iteration k of the algorithm, this search may involve solving or showing that no solution exists for up to $t_1 t_2 \cdots t_k$ linear inequality systems of the form (15) and (16), where, for each $i = 1, 2, \dots, k$, t_i is the number of extreme point neighbors in X to x^i. It is expected that the number of such linear systems (15) and (16) that must be solved will grow rapidly as m and n increase. Thus, to mitigate the computational effort of Step 3, we have incorporated Remark 3 into the tree search. In effect, the approach of Remark 3 allows for the elimination from consideration of potentially a large number of systems of the form (15) and (16).

To illustrate the Weight Set Decomposition Algorithm, we now use it to find $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ for problem (MOLP) with

$$C = \begin{bmatrix} 1 & 2 & 0 \\ -2 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 2 & 0 & 1 \\ -1 & 3 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$b = \begin{bmatrix} 6 \\ 6 \\ 2 \\ 2 \end{bmatrix}.$$

It can be shown that in this problem, the decision set X has eight extreme points, four of which are efficient, while the outcome set Y has only four extreme points, two of which are efficient. To find $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ for this problem, the Weight Set Decomposition Algorithm steps can proceed as follows.

Iteration 1.

- *Step 1.* Set $W^1 = \{w \in \mathbb{R}^2 \,|\, w > 0\}$. Choose, for instance, $w^1 = (1,1)^{\mathrm{T}} \in W^1$. Solving linear program $\mathrm{LP}(w^1)$ by the simplex method, we find the optimal extreme point solution $\bar{x}^1 = (0, 2, 0)^{\mathrm{T}}$. Set $k = 1$, $\mathrm{EX}^0 = \emptyset$ and $\mathrm{EY}^0 = \emptyset$.
- *Step 2.* The simplex method reveals that $\bar{x}^1$ is not the unique optimal solution for problem $\mathrm{LP}(w^1)$. Therefore, we use the linear programming-based procedure given in Section 3 to find $y^1 = (4,0)^{\mathrm{T}}$ and $x^1 = (0,2,0)^{\mathrm{T}}$. Using Remark 1, we find $S_{\mathrm{at}}(x^1)$ and we set $D'(1) = S_{\mathrm{at}}(x^1) - \{x^1\} = \{(0,-2,0)^{\mathrm{T}}, (0,0,2)^{\mathrm{T}}, (3,1,0)^{\mathrm{T}}\}$. We set $\mathrm{EX}^1 = \{(0,2,0)^{\mathrm{T}}\}$, $\mathrm{EY}^1 := \{(4,0)^{\mathrm{T}}\}$.
- *Step 3.*
  - *3.1.* Since $k = 1$, we set $L_1 = D'(1) = \{(0, -2, 0)^{\mathrm{T}}, (0,0,2)^{\mathrm{T}}, (3,1,0)^{\mathrm{T}}\}$ and $M = 3$. We set $j = 1$.
  - *3.2.* We choose $p^1 = ((0,-2,0)^{\mathrm{T}})$ from L_1, and we let $(\hat{x}^1) = ((0,-2,0)^{\mathrm{T}})$.
  - *3.3.* Solving (15) and (16) given by

    $$w^{\mathrm{T}} C \hat{x}^1 \geqslant 1,$$
    $$w \geqslant e$$

    shows that this linear system has no solution w. Therefore, we set $L_1 = L_1 \setminus \{p^1\} = \{(0,0,2)^{\mathrm{T}}, (3,1,0)^{\mathrm{T}}\}$. Since $j = 1 < 3 = M$, we set $j = 2$ and go to Step 3.2.
  - *3.2.* We choose $p^2 = ((0,0,2)^{\mathrm{T}})$ from L_1, and we let $(\hat{x}^1) = ((0,0,2)^{\mathrm{T}})$.
  - *3.3.* Solving (15) and (16) given by

    $$w^{\mathrm{T}} C \hat{x}^1 \geqslant 1,$$
    $$w \geqslant e$$

    shows that this linear system has no solution w. Therefore, we set $L_1 = L_1 \setminus \{p^2\} = \{(3, 1,0)^{\mathrm{T}}\}$. Since $j = 2 < 3 = M$, we set $j = 3$ and go to Step 3.2.

- *3.2.* Since $L_1 = \{(3,1,0)^T\}, p^3 = ((3,1,0)^T)$. Let $(\hat{x}^1) = ((3,1,0)^T)$.
- *3.3.* Solving (15) and (16) given by

$$w^T C\hat{x}^1 \geqslant 1,$$
$$w \geqslant e$$

now yields a feasible solution $w^* = (2,1)^T$. Therefore, we set $w^2 = (2,1)^T$ and proceed to Step 4.

- *Step 4*. Solving the linear program LP(w^2), we find the unique extreme point optimal solution $\bar{x}^2 = (3,3,0)^T$. We set $k = 2$ and go to Step 2.

Iteration 2.

- *Step 2.* Since $\bar{x}^2$ is the unique optimal solution for problem LP(w^2), we set $x^2 = \bar{x}^2 = (3,3,0)^T$ and $y^2 = C\bar{x}^2 = (9,-6)^T$. By using Remark 1, we find $S_{at}(x^2)$ and set $D'(2) = S_{at}(x^2) - \{x^2\} = \{(-3,-1,0)^T, (-1,-1/3,2)^T, (0,-3,0)^T\}$. Next, we set $\mathrm{EX}^2 = \{(0,2,0)^T, (3,3,0)^T\}$ and $\mathrm{EY}^2 = \{(4,0)^T, (9,-6)^T\}$.
- *Step 3.*
  - *3.1.* We set $L_2 = \{(p,p') \mid p \in L_1, p' \in D'(2)\}$, where $L_1 = \{(3,1,0)^T\}$. Then $M = 3$, and we set $j = 1$.
  - *3.2.* We pick $p^1 = (p,p') = ((3,1,0)^T, (-3,-1,0)^T) = (\hat{x}^1, \hat{x}^2) \in L_2$.
  - *3.3.* We find that the linear system (15) and (16) given by

$$w^T C\hat{x}^i \geqslant 1, \quad i = 1,2,$$
$$w \geqslant e$$

  is infeasible. Therefore, we let $L_2 = L_2 \setminus \{p^1\}$ and, since $j = 1 < 3 = M$, we set $j = 2$ and go to Step 3.2.
  - *3.2.* We pick $p^2 = (p,p^1) = ((3,1,0)^T, (-1,-1/3,2)^T) = (\hat{x}^1, \hat{x}^2) \in L_2$.
  - *3.3.* The system (15) and (16) with the new values for $\hat{x}^i$, $i = 1,2$, is infeasible. We set $L_2 = L_2 \setminus \{p^2\}$ and $j = 3$, and we return to Step 3.2.
  - *3.2.* We select the only remaining element p^3 in L_3, where $p^3 = ((3,1,0)^T, (0,-3,0)^T) = (\hat{x}^1, \hat{x}^2)$.
  - *3.3.* Solving the linear system (15) and (16) with the new values for $\hat{x}^i$, $i = 1,2$, shows that it is still infeasible. Therefore, we set $L_2 = L_2 \setminus \{p^3\} = \emptyset$. Since $j = 3 = M$, we stop: $\mathrm{EY}^2 = \{(4,0)^T, (9,-6)^T\} = Y_E \cap Y_{ex}$.

Notice that since $Y_E \cap Y_{ex}$ contains two elements in the sample problem, the Weight Set Decomposition Algorithm terminates after two iterations. To generate $Y_E \cap Y_{ex}$, the algorithm requires that four linear programs be solved and that six linear inequality systems of the form (15) and (16) be solved or be shown to be infeasible. An additional six linear inequality systems of the form (15) and (16) did not have to be considered due to the operations of the tree search elimination process. Finally, notice that bookkeeping is relatively simple, and no backtracking to previously generated points of $Y_E \cap Y_{ex}$ occurs.

6. Concluding observations

The Weight Set Decomposition Algorithm has a number of potential practical and computational advantages. Here we indicate some of the key potential advantages, with special attention to those not shared by other vector maximization approaches.

1. Because the Weight Set Decomposition Algorithm generates the set of all efficient extreme points in the outcome set Y rather than in the decision set X of problem (MOLP), it has all of the general potential advantages that outcome set-based vector maximization methods have over decision set-based vector maximization methods (cf. Section 1, [7,8]). Chief among these are the potential for the Weight Set Decomposition approach to be computationally less demanding than decision set-based methods and that its output will be less likely to overwhelm the decision maker than decision set-based methods.

2. The Weight Set Decomposition Algorithm is finite and exact. After q iterations, it generates all of the efficient extreme points in the outcome set Y and stops, where q is the number of efficient extreme points in Y.

3. Unlike typical decision set-based vector maximization methods for problem (MOLP), the Weight Set Decomposition Algorithm does not require any special bookkeeping, backtracking or degeneracy procedures.

4. The Weight Set Decomposition Algorithm can be implemented by using the simplex method, simplex method-like tableaus and pivots, and any method for solving linear systems.

We expect that the most demanding aspect of the algorithm computationally will involve the implementation of the global tree search procedure used in Step 3 of the algorithm. Thus, it will be especially important to implement this step in as efficient a manner as possible.

In [7,8], the global search involved in generating the set $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ is accomplished by using the outer approximation approach of global optimization in the outcome space $\mathbb{R}^p$ of problem (MOLP). In the Weight Set Decomposition Algorithm presented here, the required global search is accomplished by the tree search procedure, which involves iteratively solving linear systems of inequalities that each involve p variables. All three of these algorithms have the potential to solve larger-scale instances of problem (MOLP) by vector maximization than have heretofore been solved. Comparative computational testing will be required to ascertain which of these algorithmic approaches, if any, can best achieve this goal.

Acknowledgements

H.P. Benson gratefully acknowledges the research support provided to him by the Warrington College of Business Administration, University of Florida, Gainesville, FL.

Appendix A

Proof of Theorem 5. Since Y is nonempty and compact, problem (MOLP) has at least one efficient extreme point in the outcome set Y. As we have seen, for each $k \geqslant 1$, Step 2 finds a point $y^k \in Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$. Therefore, the number of elements q in $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ satisfies $q \geqslant 1$ and, when Step 2 is executed for the first time, it finds a point $y^1 \in Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$.

Assume now that $k \geqslant 1$ and $EY^k = \{y^1, y^2, \ldots, y^k\}$. Then, from Theorems 3 and 4, $y^i \in Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$, $i = 1, 2, \ldots, k$.

Suppose that in Step 2, $\mathrm{EY}^k \neq Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$. By setting $I = \{1, 2, \ldots, k\}$ in Theorem 2, we see that this implies that

$$W^0 \cap \left(\bigcup_{\mathrm{i}=1}^{k} W(y^i) \right)$$

is a proper subset of W^0. Therefore, there exists a weight $w \in W^0$ such that

$$w \notin \bigcup_{i=1}^{k} W(y^i),$$

i.e., there exists a weight $w \in W^{k+1}$. This implies that in Step 3, a point $w^{k+1} \in W^{k+1}$ is found by the approach, and the approach will continue to at least Step 2 of iteration $k+1$. In Step 2 of iteration $k+1$, since

$$w^{k+1} \notin \bigcup_{i=1}^{k} [W^0 \cap W(y^i)],$$

and since there is a one-to-one correspondence between the elements of $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ and the subsets $[W^0 \cap W(y^i)]$, $i = 1, 2, \ldots, q$, of W^0, an element y^{k+1} of $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ will be found that is distinct from each of y^i, $i = 1, 2, \ldots, k$.

Summarizing, we have shown that for any $k \geqslant 1$, if, in Step 2, $EY^k \neq Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ the approach will continue to iteration k+1 where it will find an unexplored point in $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$. Since the number of elements q in $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$ is finite, this implies that after q executions of Step 2, EY^q will equal $Y_{\mathrm{E}} \cap Y_{\mathrm{ex}}$, W^{q+1} will be empty, and the approach will stop. □

Proof of Lemma 1. Recall that

$$W(y) = \{w \in \mathbb{R}^p \,|\, \langle w, y \rangle \geqslant \langle w, y' \rangle \text{ for all } y' \in Y\}.$$

Therefore, since $y = Cx$, by the definition of Y,

$$W(y) = \{w \in \mathbb{R}^p \,|\, w^{\mathrm{T}} Cx \geqslant w^{\mathrm{T}} Cx' \text{ for all } x' \in X\}.$$

Since $x \in X$, this implies that $w \in W(y)$ if and only if x is an optimal solution for linear program $\mathrm{LP}(w)$. By duality theory of linear programming, it follows that $w \in W(y)$ if and only if for some $u \in \mathbb{R}^m$, (w, u) satisfies

$$C^{\mathrm{T}}w - A^{\mathrm{T}}u \leqslant 0,$$
$$u^{\mathrm{T}}(Ax - b) = 0,$$
$$x^{\mathrm{T}}(C^{\mathrm{T}}w - A^{\mathrm{T}}u) = 0,$$

and

$$u \geqslant 0.$$

Therefore, $w \in W(y)$ if and only if there exist a vector $\bar{u} \geqslant 0$ and a vector $\bar{v} \geqslant 0$ such that $(w, \bar{u}, \bar{v})$ satisfies the n equations

$$C^{\mathrm{T}}w - (A^I)^{\mathrm{T}}\bar{u} + (E^I)^{\mathrm{T}}\bar{v} = 0. \tag{A.1}$$

This implies that $w \notin W(y)$ if and only if (A.1) has no solution $(\bar{u}, \bar{v}) \geqslant 0$. From Farkas' Lemma, it follows that $w \notin W(y)$ if and only if there exists a vector $d \in \mathbb{R}^n$ such that

$$w^{\mathrm{T}}Cd > 0$$

and

$$d \in \{\bar{d} \in \mathbb{R}^n \,|\, A^I\bar{d} \leqslant 0,\ E^I\bar{d} \geqslant 0\}.$$

By definition (12) of D, the proof is complete. □

Proof of Lemma 2. First we show that $D \subseteq \mathrm{cone}(X - \{x\})$. Let $d \in D$, and let $t \in \mathbb{R}$ be any positive number. Then, from (12), $A^I(td) \leqslant 0$ and $E^I(td) \geqslant 0$. Since $Ax \leqslant b$ and $E^I x = 0$, this implies that

$$A^I(td + x) \leqslant b^I \tag{A.2}$$

and

$$E^I(td + x) \geqslant 0, \tag{A.3}$$

where b^I denotes the vector whose components are b_i, $i \in \mathrm{ID}(x)$.

For each $i \notin \mathrm{ID}(x)$, and $j \notin I^0(x)$, $A_i x < b_i$ and $x_j > 0$, respectively. Therefore, we may choose a $t > 0$ sufficiently small such that

$$A_i(td + x) < b_i \quad \text{for all } i \notin \mathrm{ID}(x),$$

$$td_j + x_j > 0 \quad \text{for all } j \notin I^0(x),$$

and (A.2) and (A.3) hold. This implies that for $t > 0$ sufficiently small, $td + x \in X$. Let us choose such a t. Then $td \in X - \{x\} \subseteq \mathrm{cone}(X - \{x\})$. Since $t > 0$ and $\mathrm{cone}(X - \{x\})$ is a cone, this implies that $d \in \mathrm{cone}(X - \{x\})$. Therefore $D \subseteq \mathrm{cone}(X - \{x\})$.

To show that $\mathrm{cone}(X - \{x\}) \subseteq D$, choose an arbitrary vector $d \in \mathrm{cone}(X - \{x\})$. Then we may choose a vector $x' \in X$ and a scalar $t > 0$ such that $d = t(x' - x)$. Since $x' \in X$, by the definitions of A^I and b^I, this implies that

$$A^I d = tA^I x' - tA^I x = tA^I x' - tb^I \leqslant tb^I - tb^I = 0. \tag{A.4}$$

Since $x' \in X$, by definition of E^{I}, this also implies that

$$E^I d = tE^I x' - tE^I x = tE^I x' - 0 \geqslant 0. \tag{A.5}$$

By (A.4) and (A.5), $d \in D$. Therefore, $\mathrm{cone}(X - \{x\}) \subseteq D$, and the proof is complete. □

References

[1] Y. Aksoy, Interactive multiple objective decision making: A bibliography (1965–1988), Management Research News 13 (1990) 1–8.

[2] P. Armand, C. Malivert, Determination of the efficient decision set in multiobjective linear programming, Journal of Optimization Theory and Applications 70 (1991) 467–489.

[3] R. Armann, Solving multiobjective programming problems by discrete representation, Optimization 20 (1989) 483–492.

[4] H.P. Benson, Finding an initial efficient extreme point for a linear multiple objective program, The Journal of the Operational Research Society 32 (1981) 495–498.

[5] H.P. Benson, Generating the efficient outcome set in multiple objective linear programs: The bi-criteria case, Acta Mathematica Vietnamica 22 (1997) 29–51.

[6] H.P. Benson, A geometrical analysis of the efficient outcome set in multiple objective convex programs with linear criterion functions, Journal of Global Optimization 6 (1995) 231–251.

[7] H.P. Benson, Hybrid approach for solving multiple-objective linear programs in outcome space, Journal of Optimization Theory and Applications 98 (1998) 17–35.

[8] H.P. Benson, An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem, Journal of Global Optimization 13 (1998) 1–24.

[9] H.P. Benson, E. Sun, Outcome space partition of the weight set in multiobjective linear programming, Journal of Optimization Theory and Applications 105 (2000) 17–36.

[10] H.P. Benson, E. Sun, Pivoting in an outcome polyhedron, Journal of Global Optimization 16 (2000) 301–323.
[11] V. Changkong, Y.Y. Haimes, Multiobjective Decision Making: Theory and Methodology, North-Holland, New York, 1983.
[12] J.L. Cohon, Multiobjective Programming and Planning, Academic Press, New York, 1978.
[13] J.P. Dauer, Analysis of the objective space in multiple objective linear programming, Journal of Mathematical Analysis and Applications 126 (1987) 579–593.
[14] J.P. Dauer, On degeneracy and collapsing in the construction of the set of objective values in a multiple objective linear program, Annals of Operations Research 47 (1993) 279–292.
[15] J.P. Dauer, Y.H. Liu, Solving multiple objective linear programs in objective space, European Journal of Operational Research 46 (1990) 350–357.
[16] J.P. Dauer, O. Saleh, Constructing the set of efficient objective values in multiple objective linear programs, European Journal of Operational Research 46 (1990) 358–365.
[17] J.G. Ecker, N.S. Hegner, I.A. Kouada, Generating all maximal efficient faces for multiple objective linear programs, Journal of Optimization Theory and Applications 30 (1980) 353–381.
[18] G.W. Evans, An overview of techniques for solving multiobjective mathematical programs, Management Science 30 (1984) 1268–1282.
[19] J.P. Evans, R.E. Steuer, Generating efficient extreme points in linear multiple objective programming: Two algorithms and computing experience, in: J.L. Cochrane, M. Zeleny (Eds.), Multiple-Criteria Decision Making, University of South Carolina, Columbia, SC, 1973, pp. 349–365.
[20] T. Gal, F. Geue, A new pivoting rule for solving various degeneracy problems, Operations Research Letters 11 (1992) 23–32.
[21] T. Gal, J. Nedoma, Multiparametric linear programming, Management Science 18 (1972) 406–422.
[22] R. Horst, On the vertex enumeration problem in cutting plane algorithms of global optimization, in: G. Fandel, H. Gehring (Eds.), Operations Research, Springer, Berlin, 1991, pp. 13–22.
[23] R. Horst, P.M. Pardalos (Eds.), Handbook of Global Optimization, Kluwer Academic Publishers, Dordecht, 1995.
[24] R. Horst, N.V. Thoai, Modification, implementation, and comparasion of three algorithms for globally solving linearly constrained convex minimization problems, Computing 42 (1989) 271–289.
[25] C.L. Hwang, A. Masud, Multiple objective decision-making methods and applications: A state of the art survey, Springer, Berlin, 1979.
[26] P. Korhonen, H. Moskowitz, J. Wallenius, Multiple criteria decision support – A review, European Journal of Operational Research 63 (1992) 361–375.
[27] D.T. Luc, Theory of Vector Optimization, Springer, Berlin, 1989.
[28] K.M. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Dordrecht, 1999.
[29] J.N. Morse, Reducing the size of the nondominated set: Pruning by clustering, Computers and Operations Research 7 (1980) 55–66.
[30] R.T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, NJ, 1970.
[31] R.E. Rosenthal, Principles of multiobjective optimization, Decision Sciences 16 (1985) 133–152.
[32] Y. Sawaragi, H. Nakayama, T. Tanino, Theory of Multiobjective Optimization, Academic Press, Orlando, FL, 1985.
[33] S. Sayin, An algorithm based on facial decomposition for finding the efficient set in multiple-objective linear programming, Operations Research Letters 19 (1996) 87–94.
[34] W. Stadler, A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960, Journal of Optimization Theory and Applications 29 (1979) 1–52.
[35] R.E. Steuer, A five-phase procedure for implementing a vector-maximum algorithm for multiple-objective linear programming problems, in: H. Thiriez, S. Zionts (Eds.), Multiple-Criteria Decision Making, Jouy-en-Josas, France, Springer, New York, 1976, pp. 159–169.
[36] R.E. Steuer, Multiple-Criteria Optimization: Theory, Computation, and Application, Wiley, New York, 1986.
[37] P.L. Yu, Multiple-Criteria Decision Making, Plenum, New York, 1985.
[38] P.L. Yu, M. Zeleny, The set of all nondominated solutions in linear cases a multicriteria simplex method, Journal of Mathematical Analysis and Applications 49 (1975) 430–468.
[39] M. Zeleny, Linear Multiobjective Programming, Springer, New York, 1974.
[40] M. Zeleny, Multiple-Criteria Decision Making, McGraw-Hill, New York, 1982.